



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

MN

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,924	02/27/2004	Craig S. Kickel	303.851US1	8884
21186	7590	07/16/2007	EXAMINER	
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.			LUDWIG, MATTHEW J.	
P.O. BOX 2938			ART UNIT	PAPER NUMBER
MINNEAPOLIS, MN 55402			2178	
			MAIL DATE	DELIVERY MODE
			07/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/789,924	KICKEL, CRAIG S.	
	Examiner	Art Unit	
	Matthew J. Ludwig	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 16 April 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-9, 11-37, 39-42 and 44-56 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-9, 11-37, 39-42, and 44-56 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date: _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date: _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. This action is in response to the Request for Continued Examination received 4/16/2007.
2. Claims 1-4, 5-9, 11-37, 39-42, and 44-56, are pending in the application.
3. Claims 1-4, 5-9, 11-37, 39-42, and 44-56, rejected under 35 U.S.C. 103(a) as being unpatentable over Tijare in view of Lipkin have been withdrawn pursuant to applicant's amendment. Claims 22-27 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter have been withdrawn pursuant to applicant's amendment.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.
5. **Claims 1-4, 6-9, 11-15, are rejected under 35 U.S.C. 103(a) as being unpatentable over Periwal USPN 6,163,776 filed (3/23/1998).**

In reference to independent claim 1, Periwal teaches:

The schema generator is a tool or routines for controlling the processor to generate the relational database schema (table definitions, constraint definitions) as well as metadata information corresponding to the object class definitions and the ORM data structures during initialization (compare to "*reading metadata for a database, the metadata including a constraint definition for at least one column of the database*""). See column 7, lines 1-16. The

reference specifies commands that can be used to store the ORM Specification in the database, where the ORMid identifies the specification uniquely. The text of the ORM File goes into the MetaInfo field, and the MetaFileName field is initialized with the ORM File name for recording purpose. Even if the RDBMS does not support a LONGVARCHAR style field, it is easy to store the MetaInfo text in multiple rows of a separate table with a varchar field and an ORMid field. See column 7, lines 50-67 and column 8, lines 1-15. The reference fails to explicitly state one column of a database; however, the reference does mention storing tables and their primary keys, foreign keys for a database. It would have been obvious to one of ordinary skill in the art, having the well-known database methods to have modified the storage techniques and utilized distinct columns of the database because it would have further defined the organization and structure of the data in the relational model.

The database exchange unit executes in accordance with the ORM specification, and is the program/routines that operate to translate data from the object model to the relational model, and vice versa. The schema reverse engineering unit creates Object Class Definitions and a database schema. The schema generator generates the RDBMS schema from Object Class Definitions and the ORM specification. See column 6, lines 45-67. Furthermore, the memory also includes applications and a text editor or graphical user interface. The applications may be any one of a conventional type written in object-oriented languages such as Java. The applications are routines or modules that allow the user to access data in the system (compare to *“translating the metadata into a page description language, the page description language including a translation in accordance with the constraint definition, and outputting the page description language”*).

In reference to dependent claim 2, Periwal teaches:

The schema generator is also stored in memory. The Schema Generator is a tool or routines for controlling the processor to generate the relational database schema (table definitions, constraint definitions) as well as metadata information corresponding to the object class definitions and the ORM data structures during initialization. After the relationship database schema has been created they are stored in memory or the RDBMS and used by OR Mapping Unit to map object calls to database requests. See column 7, lines 1-43.

In reference to dependent claim 3, Periwal teaches:

The reference specifies commands that can be used to store the ORM Specification in the database, where the ORMid identifies the specification uniquely. The text of the ORM File goes into the MetaInfo field, and the MetaFileName field is initialized with the ORM File name for recording purpose. Even if the RDBMS does not support a LONGVARCHAR style field, it is easy to store the MetaInfo text in multiple rows of a separate table with a varchar field and an ORMid field. See column 7, lines 50-67 and column 8, lines 1-15. The reference fails to explicitly state one column of a database; however, the reference does mention storing tables and their primary keys, foreign keys for a database. It would have been obvious to one of ordinary skill in the art, having the well-known database methods to have modified the storage techniques and utilized distinct columns of the database because it would have further defined the organization and structure of the data in the relational model.

In reference to dependent claim 4, Periwal teaches:

Finally, the method determines whether there are more non-null referenced objects to be inserted. If there are more non-null referenced objects to be inserted, the method returns to step to create an insert statement for each of them. See column 18, lines 1-13.

In reference to dependent claim 6, Periwal teaches:

Included in memory is a conventional text editor or graphical user interface. The text editor or graphical user interface may be any one of a number of conventional editing interfaces may be any one of a number of conventional editing interfaces. Since the ORM specification is a text file, any type of text editors or graphical user interfaces may be included in the system. See column 8, lines 25-38.

In reference to dependent claim 7, Periwal teaches:

The specification of mapping information using this innovative grammar allows the mapping information to be stored conveniently and easily in an operating system file and also as part of the relational database. The present invention also provides an application programming interface (API) that automates the tedious object-oriented program translation code. See column 3, lines 10-25.

In reference to dependent claim 8, Periwal teaches:

The schema reverse engineering unit is a routine or tool for creating object class definitions and ORM specification using a database schema (table definitions, constraint definitions) for a set of tables. The schema reverse engineering unit provides the user, given a relational model, with the ability to generate the IRM specification for translating between the given relational model and an object-oriented model. See column 8, lines 39-49.

In reference to dependent claim 9, Periwal teaches:

This schema generator takes the name of the file containing the object-relational mapping information as input, and generates three script files containing the sql ddl statements for creating the required tables and their primary keys; foreign key constraints. See column 7, lines 13-29.

In reference to dependent claim 11 and 12, Periwal teaches:

The memory also includes a Schema Reverse Engineering Unit. The schema reverse engineering unit is a routine or tool for creating object class definitions and ORM specification using a database schema (table definitions, constraint definitions) for a set of tables. The schema reverse engineering unit provides the user, given a relational model, with the ability to generate the ORM specification for translating between the given relational model and an object-oriented model. The schema reverse engineering unit also generates or defines object-oriented model by creating the object class definitions. See column 8, lines 38-67.

In reference to dependent claim 13, Periwal teaches:

The object-relational mapping system of the present invention provides for the automatic and systematic exchange of data and commands between an object-oriented system and a relational system. See column 5, lines 1-40.

In reference to dependent claim 14, Periwal teaches:

The object-relational mapping system of the present invention provides for the automatic and systematic exchange of data and commands between an object-oriented system and a relational system. See column 5, lines 1-40.

In reference to dependent claim 15, Periwal teaches:

The schema generator takes the name of the file containing the object-relational mapping information as input, and generates three script files containing the SQL DDL statements. The reference fails to explicitly state a coldfusion scripting language, however, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the well known scripting languages as taught by Periwal and provided a user with the additional coldfusion scripting language because it would have provided multiple scripting languages in the translating/exchanging of data.

6. Claims 16-42 and 44-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Periwal in view of Fuller et al., USPN 6,877,134 filed (7/29/1999).

In reference to dependent claim 16-18, Periwal teaches:

Exchanging data and commands between an object oriented system and a relational database. Languages and programming paradigms are used in conjunction with data that can be stored persistently in a variety of different ways. See column 1, lines 10-54. The language found in the reference fails to explicitly state metadata translated into HTML, however, Fuller allows a user to access metadata and process it into HTML form in a browser window, which also involves keyframe images in a keyframe frame. Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Periwal and Fuller before them at the time the invention was made to provide a user with metadata and capturing the metadata in HTML form as Periwal reference different language make use of the exchange of data for efficient content retrieval.

In reference to independent claim 19 and 20, Periwal teaches:

The schema generator is a tool or routines for controlling the processor to generate the relational database schema (table definitions, constraint definitions) as well as metadata information corresponding to the object class definitions and the ORM data structures during initialization (compare to “*reading metadata for a database, the metadata including a constraint definition for at least one column of the database*”). See column 7, lines 1-16. The reference specifies commands that can be used to store the ORM Specification in the database, where the ORMid identifies the specification uniquely. The text of the ORM File goes into the MetaInfo field, and the MetaFileName field is initialized with the ORM File name for recording purpose. Even if the RDBMS does not support a LONGVARCHAR style field, it is easy to store the MetaInfo text in multiple rows of a separate table with a varchar field and an ORMid field. See column 7, lines 50-67 and column 8, lines 1-15. The reference fails to explicitly state one column of a database; however, the reference does mention storing tables and their primary keys, foreign keys for a database. It would have been obvious to one of ordinary skill in the art, having the well-known database methods to have modified the storage techniques and utilized distinct columns of the database because it would have further defined the organization and structure of the data in the relational model.

The database exchange unit executes in accordance with the ORM specification, and is the program/routines that operate to translate data from the object model to the relational model, and vice versa. The schema reverse engineering unit creates Object Class Definitions and a database schema. The schema generator generates the RDBMS schema from Object Class Definitions and the ORM specification. See column 6, lines 45-67. Furthermore, the memory

also includes applications and a text editor or graphical user interface. The applications may be any one of a conventional type written in object-oriented languages such as Java. The applications are routines or modules that allow the user to access data in the system (compare to “*translating the metadata into a page description language, the page description language including a translation in accordance with the constraint definition, and outputting the page description language*”).

Exchanging data and commands between an object oriented system and a relational database. Languages and programming paradigms are used in conjunction with data that can be stored persistently in a variety of different ways. See column 1, lines 10-54. The language found in the reference fails to explicitly state metadata translated into HTML, however, Fuller allows a user to access metadata and process it into HTML form in a browser window, which also involves keyframe images in a keyframe frame. Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Periwal and Fuller before them at the time the invention was made to provide a user with metadata and capturing the metadata in HTML form as Periwal reference different language make use of the exchange of data for efficient content retrieval.

In reference to dependent claim 21, Periwal teaches:

The object-relational mapping system of the present invention provides for the automatic and systematic exchange of data and commands between an object-oriented system and a relational system. See column 5, lines 1-40.

In reference to claims 22-42, 44-56, the claims reflect similar limitations to those claims rejected in 1-21, and therefore, the claims are rejected under similar rationale.

Response to Arguments

7. Applicant's arguments with respect to claims 1-9, 11-37, 39-42, and 44-56 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew J. Ludwig whose telephone number is 571-272-4127. The examiner can normally be reached on 9:00am-6:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



STEPHEN HONG

EXAMINER

ML